

VideoCD Player for CD32

I worked on several modules of the CD32. Among those projects was revising the CD audio player from CDTV. We had a new visual design from Jim Sachs, and I had to bring that to life and make some minor adjustments. Also I made various bug fixes and some small enhancements.

Once the CD32 itself was complete, attention turned to the MPEG module for the CD32. We were planning to support the VideoCD format, which allowed 72 minutes of MPEG-1 video and audio on a disc, for music videos, karaoke CDs, and movies. I did the design for the VideoCD player user-interface in August of 1993, shortly before I left Commodore. I present here the design description and mockups, as well as other text notes and visual artifacts.

Monospace Bold text is the filename of the image / text being presented.

Any text in green italic, like this, is modern commentary.

Abbreviations and names:

- Martin / MT: Martin Taillefer, developer
- Ken / KCD: Ken Dyke, developer
- Darren / DG: Darren Greenwald, developer
- JH: Jerry Horanoff, developer

Peter Cherna
March, 2020

VideoCD allows 72 minutes of MPEG video/audio to be encoded onto a regular CD-ROM. This format will be used for Karaoke CDs, music video CDs, as well as movies (much like VHS). The spec governing this is the Karaoke-CD spec, which provides for different kinds of information to be stored on the disc, such as:

- * Disc title
- Disc contents description (similar what is printed on the disc jacket)
- List of tracks
- For each track:
 - * Track title
 - * Performer's name
 - * Songwriter's name
 - * Composer's name
- Song text

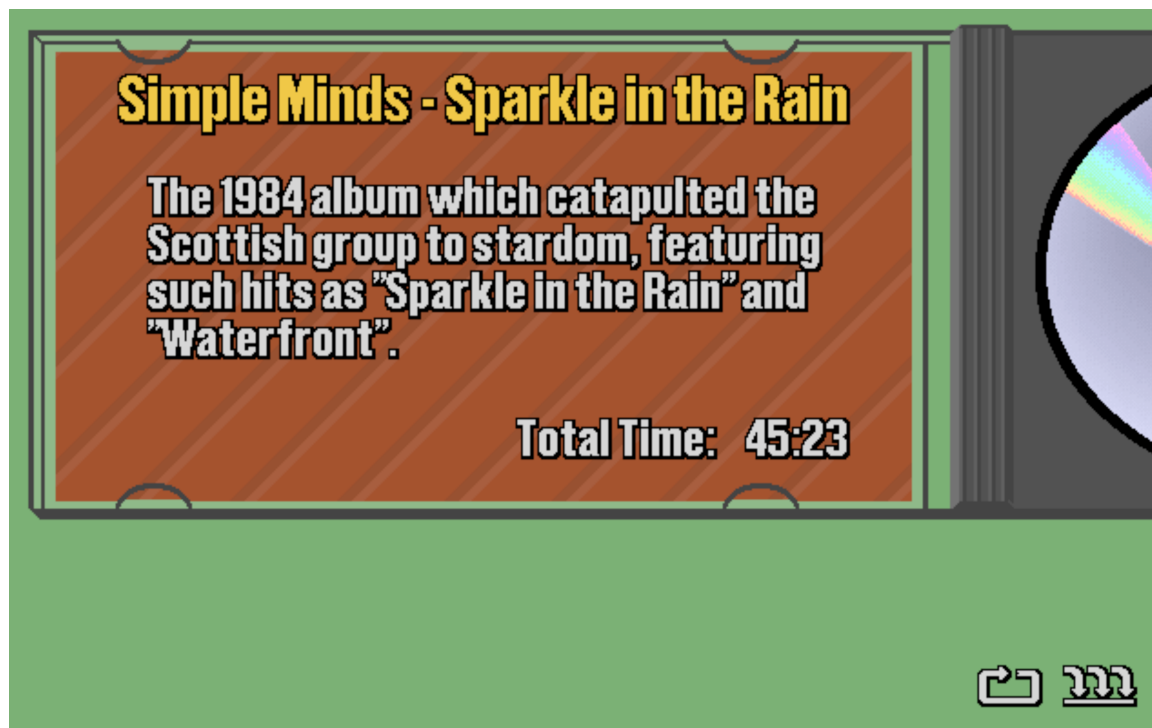
(* indicates a required item, while - indicates optional items).

The basic idea of the user-interface is to present the user with an open CD jewel-box (the standard CD plastic case). The booklet insert occupies most of the screen, and all the browsing and most of the selection takes place on different pages of the booklet.

The picture file is actually composed as an ANIM having five frames, with each frame representing one of the booklet pages presented by the system for a certain music video CD. We anticipate some kind of page-turning animation to move from page to page.

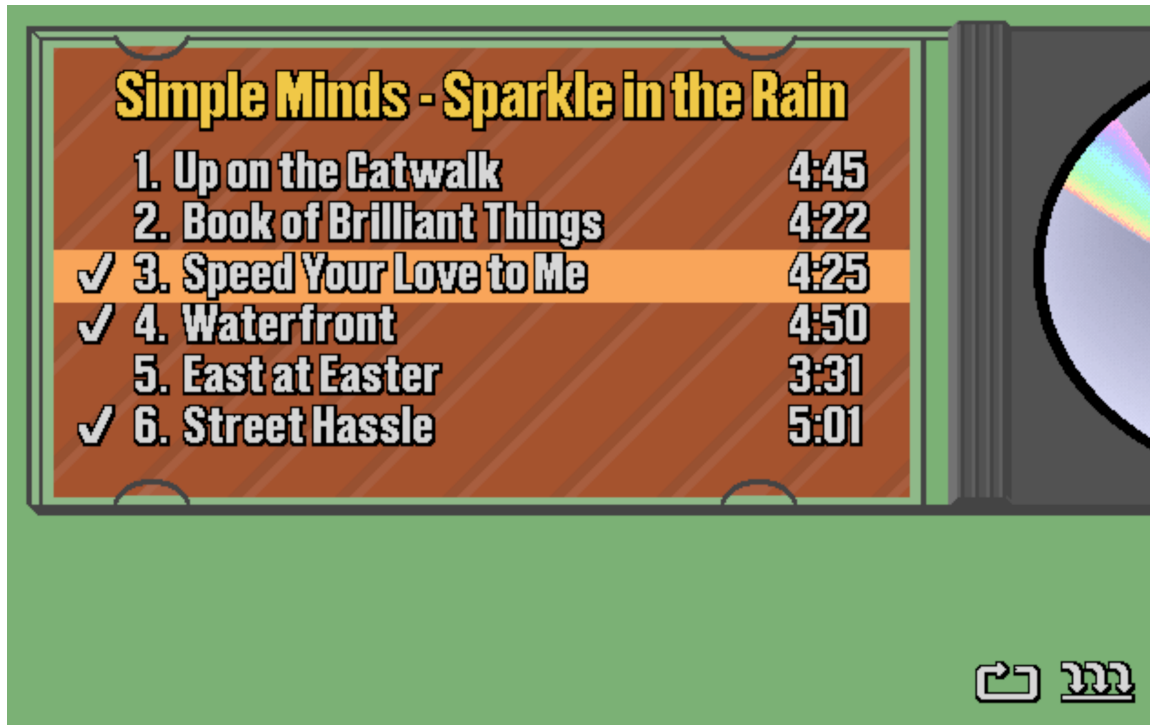
When I originally shared the document internally, I attached a file JewelBox.anim, whose five frames are shown below. The previous paragraph is talking about that anim file.

Frame 1:



This frame presents the disc contents description and disc title, as well as the total playing time. If the disc contents description was absent, this page would be omitted. The playing time may move into a separate clock, to be invented later.

Frame 2:



This frame is the most important one, as it presents the track selection list beneath the disc title. A checkmark indicates that a track has been selected for play, while the bright highlight bar indicates the "current" track. Moving through the list moves the bright bar, and pressing the CD32 fire button toggles the checkmark. We have three options for scrolling the list of tracks. In one option, the booklet "turns" to the next page to reveal the next set of tracks. In another option, the track list scrolls on this page when the bar can go no further up or down. In the third option, the bar is fixed, and the list scrolls "behind it" (in the style of the language and NV RAM screens already in the CD32 ROM).

Frame 3:



This frame holds whatever per-track information (other than lyrics) which is available and deemed desirable for display. In this case, we show the performer's and songwriter's names. We expect to be able to grab and scale frames of video, so we include a "representative" still frame from the current track for extra allure.

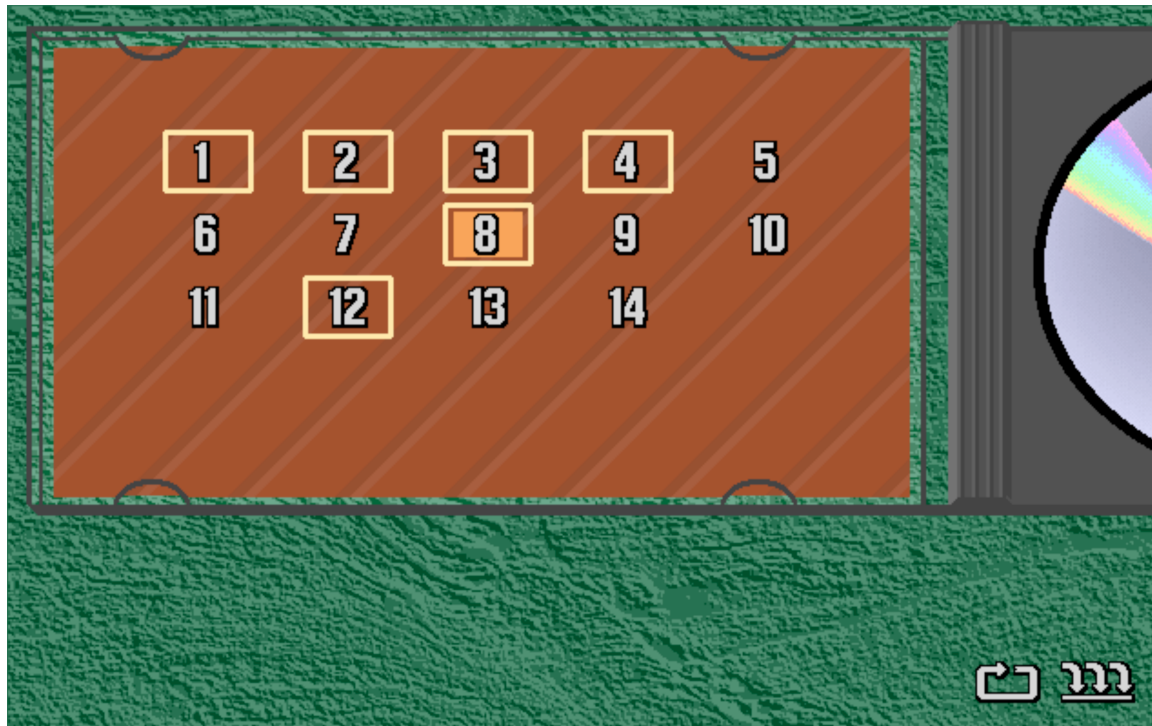
The still image sample I used was a black-and-white scan of three of Commodore's developers (Jerry Horanoff, Spencer Shanson, and myself), I think taken at the Orlando DevCon in 1993.

Frame 4:



This frame holds the lyrics for the song, if available. If the lyrics do not fit on a single page, we will either have multiple pages or we will allow the lyrics to scroll on one page. If lyrics are not available, this frame is omitted.

Frame 5:



This frame is not part of the VideoCD sequence, but represents a quick design of what we might present if a regular audio CD is inserted for play. The design is tricky because audio CDs have no information beyond track numbers and track lengths, so there is not much information to present. We may opt to keep the current in-ROM audio player, since it's more visually interesting than this design (and it's complete), but if a reasonable design can be found, consistency with VideoCD would be desirable. This screen is currently the weakest of the five in design.

The little yellow left and right arrows are supposed to indicate that the user can turn to other pages by pressing the controller's joypad in that direction.

The loop (restart the playing of the selection when it ends) and sample-mode (play only 10 seconds of each track) indicators at the bottom-right of the screen are used to denote the state of those toggle options. A shuffle mode glyph would be added for that option too.

For audio CDs, I had the idea that we'd have ten or sixteen different color palettes, and then use a hash of the audio-CD table of contents, so that a given CD was always use the same color scheme, even though which scheme chosen would be random.

There are two major elements which are completely absent at this time. First, there needs to be some kind of clock-indicator which might display the cumulative duration of the selected tracks. We played very briefly with analog, digital, and hybrid designs, but have not found a design we like. The other major element which is absent is some kind of pop-up menu, in which options such as sample mode, select audio track, show song-text over video are selectable. Possibly, the menu would be implemented on another "page" of the booklet.

The following are design parameters which need to be respected:

- The way it is currently drawn, the CD box is too wide compared to its height. An actual jewel box is a bit wider than it is tall, which we can perhaps take advantage of in order to share artwork between NTSC and PAL -- say having it nearly square in NTSC and a bit too wide when in PAL).
- The current picture does not use overscan. The final artwork should fill the horizontal width of the monitor. We could opt to use the full vertical height as well, but for technical reasons it may be desirable for the background pattern to fade to a solid color at the top and bottom, thus we would not have actual pixels for the maximum overscan height, but rather just the nominal 400 lines.
- If possible, the final picture should use no more than 128 colors. Some of the open issues will require color registers to implement (e.g., a menu or a representation of the controller as a help guide). We could potentially use all 256 colors, but there will be a ROM, RAM, and performance impact for that. A question would be how much would the image suffer from restricting it to a lesser number of colors?
- We need to use independent palette entries for each "use" of color. For example, the black text-outline should be a different pen from black used in the jewel box or the background, even if it is exactly the same hue. We need this to preserve flexibility. Also, colors should be nicely grouped, e.g., one consecutive range of colors for the background, one for the booklet color, one for the CD rainbow, etc.
- We may need to increase the inter-character spacing of the font a bit for legibility.
- We need to have room for at least 6 lines of text beneath the title. (Maybe 5 in a pinch). (Currently, we're using Scala's Compact 36 and 24 for title and text respectively).
- We would like to have multiple color sets and/or patterns for the booklet paper (which is striped deep red in the example). We can then arrange for different color/pattern combinations for different discs.

- An MPEG picture is 352x240 pixels, lores/non-lace. The sample picture on the third frame was chosen to be exactly half that size. Since our screen is hires/lace, the result is a picture one quarter the width/height of the video screen. It may be desirable to keep a simple scaling relationship between the original video and the reduced-size picture.

The following are open questions which eventually need to be answered:

- Should the representation of the CD, jewel box, and surface be abstract/cartoon/stylized, or photorealistic?
- We currently use a checkmark to denote tracks which are selected for play. Could we use some kind of highlight instead? Compare to the audio-CD screen (Frame 5), for which a highlight is preferable to a checkmark.
- On the track-selection page (Frame 2), should we scroll the highlight bar, or scroll the list behind a fixed bar?
- Should we include a representation of the CD32 controller somehow illustrating the function of most of the buttons?
- We need to invent some kind of cue to invite the user to scroll up and down through the track-selection list when there are more tracks to be seen (akin to the yellow arrows which indicate that there are additional pages to turn to).
- Almost all of the text shown actually comes off the VideoCD. The only text shown which does not are the words "Total Time" on Frame 1 and the phrases "Written by" and "Performed by" on Frame 3. The "Total Time" would probably disappear when we finalized the clock design. The other two would either need to be localized, or turned into pictograms. Can meaningful icons for these phrases be invented?
- We could add other pictograms to help identify each page. The first frame, which describes the CD as a whole, might have a CD pictogram on it. The lyrics page could use some other kind of pictogram as well.

What follows are various text files with related notes, and various visual assets, some of which are alternate design concepts, and some of which are "ingredients" for building the other imagery.

JewelBox.notes

Blit size, 250x100

DEPTH	OVERSCAN	BLIT SPEED	BLIT PERFORMANCE RATIO
8 bitplanes	OSCAN_MAX	34.5 blits/sec	1.00
8 bitplanes	OSCAN_MIXED	39.3 blits/sec	1.14
8 bitplanes	OSCAN_TEXT	42.9 blits/sec	1.24
7 bitplanes	OSCAN_MAX	43.2 blits/sec	1.25
7 bitplanes	OSCAN_MIXED	48.0 blits/sec	1.39
7 bitplanes	OSCAN_TEXT	51.0 blits/sec	1.48

- Should the representation of the CD, jewel box, and surface be abstract/cartoon/stylized, or photorealistic?
- The CD box is too wide. (NB: an actual jewel box is a bit wider than it is tall, which we'll have to take advantage of in order to share artwork between NTSC and PAL)
- Make picture overscan
- Determine number of colors we can afford to spend (depends on bandwidth, number of colors needed for special effects, etc.)
- Need to use independent palette entries for each type of thing. For example, black text-outline should be a different pen from black used in the jewel box or the background, even if the same hue. We need this to preserve flexibility. Also, colors should be nicely grouped, e.g., a consecutive range of colors for the background, one for the booklet color, one for the CD rainbow.
- Checkmark or highlight for "selected for play"?
- Should we scroll the highlight bar, or scroll the list behind a fixed bar?
- On-screen help guide (a representation of the controller)?
- Increase font inter-character spacing
- Cues to scroll up/down in the list
- * Cues to page left/right in the booklet
- Could select multiple color sets for the booklet, then perform some hashing function over the disc ID, so we get random but repeatable coloring. (Could do for background too)
- Mouse use?
- Picture is exactly half-size
- "Written By" "Performed By": localize or iconify?
- Frames
 - 1: "CD Jacket": contains disc title and "liner notes" if any
 - 2: "Track Select": contains list of tracks
 - 3: "Track Info": contains author/performer info, MPEG still framei
 - 4: "Song Text": contains song text if present
 - 5: "Audio": what the same player might look like for an audio disc (since no info is available)

Here are a list of the operations we hope to support, and the manner in which they're accessed:

Play

- Tap PLAY/PAUSE button while paused or stopped.

Pause

- Tap PLAY/PAUSE button while playing.

Stop

- Tap STOP button at any time.

Fast Forward

- Hold FFWD button when playing.

Rewind

- Hold FREV button when playing.

Next Track

- Tap FFWD button when playing.

Previous Track

- Tap FREV button when playing.

Single Step Forward

- Tap FFWD button when paused (may be obscure).

Single Step Backward

- Tap FREV button when paused (may be obscure).

Slow Play

- Hold FFWD (FREV?) when paused (may be obscure).

Show Title/Artist/Time over video

- Press "up/down" button to cycle through
 - title/artist, elapsed-time, state-icons
 - title/artist, remaining-time, state-icons
 - overlay off

Shuffle Play

- Tap SHUFFLE button

Loop Play

- Tap LOOP button

Menu Up

- Tap FFWD/FREV simultaneously when menu is down

Menu Down

- Tap FFWD/FREV simultaneously when menu is up

Custom track program

- FF/RW/up/down/fire while on the selection screen

Mute

- In menu

Sample Play

- In menu

Select Audio "Channel"

- In menu

Show song-text over video

- On/off is in menu. Scrolling of text (since it's not synchronized) is achieved by using the left and right keys.

Go to time in track

- In menu

Bookmark

- Need to update bookmark if you stop in the middle of a track. Should also update bookmark while playing, so you can power down and still resume. One resume strategy: if booted with the same disc in place, auto-play where left off, else reset bookmark.

Visual cueing (locate part of video by thumbnail)

- Not likely

UI brightness/hue

- Not likely

KaraokeCDSpec.problems

Synchronization for "song text", which could be used for Karaoke, subtitling/closed captioning, and other follow-along stuff. Multiple languages of subtitling would be cool.

How to find a good "representative still frame" in the MPEG stream, for each track.

Eliminate 3 seconds of blackness before each track, to allow seamless chapter marks.

Store MPEG-still encoded movie poster or album cover, at full screen, and 1/4 screen.

Ability to have a CD-single, which plays CD audio on a CD audio player, but uses the rest of the CD for a VideoCD video clip.

Menu.notes

Mute (symbol lips w/ finger)

- In menu. Toggle select.

Sample Play (existing symbol)

- In menu. If playing, toggle. If stopped, turn on and start playing.

Select Audio "Channel"

- In menu

Show song-text over video

- On/off is in menu. Scrolling of text (since it's not synchronized) is achieved by using the left and right keys.

Go to time in track

- In menu

MPEG.design

When you stick in a Music Video Disc, you'll be presented with a song list, e.g.,

Bonnie Raitt - Nine Lives		
1.	No way to Treat a Lady	3:51
2.	Runnin' Back to Me	4:14
3.	Who But a Fool (Thief into Paradise)	4:26
4.	Crime of Passion	4:20
5.	All Day, All Night	4:04
6.	Stand Up to the Night	4:43
7.	Excited	3:12
8.	Freezin' (for a Little Human Love)	4:59
9.	True Love is Hard to Find	4:34
10.	Angel	4:00

Playing Time: 42:27

The problem is that if you stick in a movie disc with no chapters, you would see a pretty disappointing display:

Star Wars Disc 1		
1.	The film	57:10
Playing Time:		57:10

A chaptered disc might be:

Star Wars Disc 1		
1.	Opening Titles	2:53
2.	Tatooine	10:00
3.	Flight of the Falcon	12:43
4.	Death Star	14:23
5.	Final Battle	9:57
6.	Celebration	1:49
7.	Credits	5:11
Playing Time:		57:10

Operations to perform when track-selector is up:

Play:

Press play key.

Track select:

Move through tracks with FF/RW with SELECT to program.
UP and DOWN also move through the track list. STOP button
clears selection. SELECT would toggle state (it could
alternately select again for repeated play of tracks).

Main Menu:

Press the MENU key to bring up the main menu. Use UP and DOWN
to move through the menu, SELECT to pick an action. MENU
brings down the main menu with no action taken.

Shuffle:

Do we need this for video? For audio?

In the main menu, select "Shuffle Play" to shuffle and play.

Do we scramble the track table when shuffled? Maybe not...

Loop Mode:

Do we need this for video? For audio?

In the main menu, select "Loop at End" to toggle loop mode.

Intro Mode:

In the main menu, select "Intro Play" to toggle loop mode.

Video Sample:

In the main menu, select "Video Sample" to display the video thumbnail samples.

Time Mode:

In the main menu, select "Time..." to bring up the TIME MENU.

The time menu contains:

Time Display Off

Track Elapsed Time

Track Remaining Time

Disc Elapsed Time

Disc Remaining Time

Use UP and DOWN to move through the choices, SELECT to pick the one you want.

ALTERNATELY:

Use the yellow button to bring up all display elements, in the order:

1. Title/Artist Track elapsed
2. Title/Artist -Track remaining
3. Title/Artist -Disc remaining
4. Off

Preferences:

In menu

MPEG.notes

How the screen should differ from the latest mockup:

- * If we have a select-by-title screen as well as a select-by-thumbnail screen, we should come up with an icon for each screen. The icon would be used in the menu (done) as well as in the upper left corner of the matching screen (not done).
- "List Pick" is a menu option in the menu brought up when playing
- Going to ListPick or VideoPick when playing stops the play but keeps the current checkmark-set.
- * In order to allow the menus to have both icons and checkmarks, we'll have to place the menu checkmarks on the right. This means we may have to move the track checkmarks too.
- * We could add some simple on-screen help, showing a mini-controller or at least just the button colors. The pop-up menu could "zoom" from the green button icon.
- In the main window, have little status icons to indicate which of loop and sample mode (and ...?) are currently selected.
- Interlace is OK. NB: It may be difficult/impossible to record off a non-interlace screen. Going NI would be like copy protection, going interlace would be a feature over the other NI implementations.
- Need pause and variable slow-play
- * Increase font spacing (RastPort->TxSpacing)
- Cue that there is more on the list that needs to be scrolled into view.
- Change menu to "Shuffle Play", "Loop Play", "Intro Play" ("Scan Play"?)
- Add "Play" to menu
- Need Slow-play, step modes. How?
 - Step mode could be FF/RW when paused. Slow play could be achieved by holding play while playing or paused (vs. tapping).
- Add help to screen
- There exist multiple audio tracks (eg. with/without Vocals) and we need to be able to control it
- Could control brightness of UI elements
- Need to do a ROM budget

Features of Karaoke-CD (* = every disc has this, o = not every disc does)

Darren mentioned something about two audio tracks (eg. one for music only, and the other having vocals (or maybe music+vocals))

Karaoke CD's have these things of interest:

- * Disc title
- o Disc contents description
 - similar to what's printed on the disc jacket

Each sequence has:

- * Title
 - title of the AV sequence
- * Performer's name
- * Songwriter's name
- * Composer's name
- o Arranger's name
- o Player's name
- o Song text
- o Karaoke key (e.g., C-sharp is "C#", B-flat-minor is "Bbm")
- o Original key
- o Description of contents of sequence

I don't see where in the spec the song text is keyed to part of the video!

Issues:

- The current CDGS audio player has a few key problems
 - Its use of a travelling cursor is confusing. There are divergent ways of navigating the tracks, for example (move cursor into track table or use FF/RW to move "highlight").
 - All game-controller buttons are "used", but the MPEG player may demand a lot of new stuff.

Here is a potential feature-set for the MPEG player:

- Existing audio functions easily supported:
 - Play, pause, stop
 - Track ahead, track back
 - Intro mode
 - Repeat mode
 - Time counter and time modes
 - Shuffle
 - Custom track programming of some sort
- Existing audio functions hard to support:
 - Fast forward, fast reverse [see notes below]
- Show disc name
- Show disc contents description
- View list of titles
- For each title, see supplementary info (performer, songwriter, etc.)
- Enable/disable display of Karaoke text as the song plays
- A cycle-menu button which goes through the function choices
- A cycle-display button which goes through the possible display info:
 - song title, artist, time, etc.
- Visual track table (shows a frame from each track) [see notes below]
- Visual track browser (shows still frames from different parts of the track, allows locating the exact place you want) [see notes below]
- BookMarks - remembers how far into the video you got, so you can re-insert the disc and resume playing from the last spot. Could remember one bookmark per disc for multiple discs.
- Audio-only mode for MPEG discs?

Issues surrounding FF/RW:

With MPEG, you can't just accelerate the data rate (like regular cassette tapes), nor can you just scan part of every frame (like a VCR), nor can you easily take time-snippets (like an audio CD).

To skip ahead and grab the next frame of video requires seeking, the settle delay, reading enough data to be sure to have an MPEG "I-frame", (key frame), then filling up the chip. We guesstimate this will take somewhere in the range of 1.5 - 3 seconds. This puts severe restrictions on the types of fast-forward we can support.

Here are two suggestions:

1. Show successive stills, changing as fast as skip-ahead will permit (say every 2 seconds for sake of argument), skipping as far ahead as needed to meet the fast-forward rate desired (e.g. 10x). In this case, we would display a frame, and immediately begin a seek to 20 seconds farther into the video. The result will be frame 0:00:00 displayed for 2 seconds, followed by frame 0:20:00 displayed for 2 seconds, then 0:40:00 for 2 seconds, etc.

2. Once we get the still up, we're substantially on the way to being able to display some actual video. So we could show (say) .5 seconds of motion before skipping ahead. The result would be .5 seconds of video, then freeze on the last frame for about 2 seconds while we seek, then .5 seconds of video from about 20 seconds further on, etc.

One big problem is that neither one of these fast-forward strategies is likely to appear to be fast. I'm sure that a visual aid could be added to help. One idea would be to overlay a "progress meter", which would be a stripe growing from left to right as the 2 seconds of static image proceeds.

A great firmware feature would be the ability to seek a specified amount, but not care exactly where you ended up, which might save the delay of fine-positioning. That is, if you want to skip 20 seconds into the video, but the head lands 19.5 seconds ahead instead, you don't really care for it to correct that error. You'd rather get the next image out as fast as possible. This might increase the rate at which we can bring up distinct frames.

Issues for visual browser:

The visual browser would display some number (e.g, 4, 9, 16) of "thumbnail" representations of stills from the video. In order to be able to display these, we need the ability to read out of the MPEG display buffer, which is not currently possible but C-Cubed has said they will do this for us. There is also a palette-sharing issue, which could be solved by showing the thumbnails in monochrome.

One problem is assuring that the sampled frame is "interesting". It would seem to be too easy to grab an all black frame. Perhaps some simple low-contrast/low-brightness criterion could reject some of these frames (at a further speed penalty, of course).

One incarnation of the browser would show a single frame from each track. This could be used for track selection.

Another incarnation of the browser would show several frames (say 16) from the movie. It would be possible to zoom in temporally, that is to say pick a frame and see 16 frames in the vicinity of the selected frame. A few successive zooms and the user would hopefully be able to locate the desired point to play from.

The speed of sampling of the thumbnails is a significant concern. It would likely be in the 1.5-3 second range per frame, due to seek/settle delays. This may be too slow to make this feature useful at all.

Scheduling Issues:

We will need some time in the schedule for prototyping, in order to answer questions about the impact of the slow still-frame retrieval rate on possible solutions for fast-forward/reverse and browsing.

As well, we need some time to do some user-testing on the design.

Another observation is that we likely will have more text in the interface of the MPEG player (akin to the on-screen controls of a TV or VCR), and that would necessitate a translation pass in the schedule.

Free buttons:

up,down,left,right,select,shuffle,loop

List of titles

Visual list of titles

Play

- Hit PLAY/PAUSE button when paused or stopped

Pause

- Hit PLAY/PAUSE button when playing

Single Step

- Tap FF/RW when paused?

Slow Play

- Hold play/pause

Stop

- Hit STOP button at any time.

Track Fwd

- Hit FFWD button when playing (or paused? -- see single step)

Track Back

- Hit FREV button when playing (or paused? -- see single step)

Fast forward

- Hold FFWD button when playing or paused.

Rewind

- Hold FREV button when playing or paused.

Intro Mode

- Menu command

Repeat Mode

- Menu command

Shuffle

- Menu command

Custom Track programming

- Menu command

Display

- time (different modes)
- disc name
- disc contents
- per-track info (performer, songwriter, etc.)
- Karaoke text (and singing keys)

Make bookmark

Clear bookmark

Define REPEAT button as MENU button

Define SHUFFLE button as DISPLAY button

MENU BUTTON

- Brings up Menu if not up, dismisses it if up.
 - Menus could have numbers if a computer-module (i.e. keyboard) is attached
- UP and DOWN in Menu move highlight between:
- Intro Mode
 - Repeat Mode
 - Shuffle Mode
 - Track select
 - ? Time mode
 - ? Other display options
 - ? Left/Right volume

DISPLAY button changes displayed info

- Time

Idea: Different preset preferences for menu style

- Some high-tech chrome font against red lucite (superhires dither)
- Some cartoon character with a pointer stick (in the style of a professor at a blackboard)
- "Large friendly letters"

MPEG.schedule

HW DEVICE DRIVER

Finalize Device API	1w	KCD
Complete Device Primitives	1w	KCD
2.0 MicroCode from CCube	15-Aug-93	Ext Milestone (CCube)
Integrate/Verify 2.0 uCode	2w	KCD
Frame Capture from Streams	1w	KCD (depends on uCode)
Zorro II device driver version	1w	KCD

HW ISSUE:

Ken's MPEG system seems unreliable when the butt-board is attached. The CL450 sometimes fails to get properly initialized, and if it does, trash always appears in the CL450's video buffer. This does interfere with his ability to work, and may also be masking real bugs in the system. It also may be foretelling potential problems when MPEG plus a computer module are attached. We should get HW to provide Ken with a stable platform, whatever it takes. We will need at least one if not two other stable platforms (for Peter and Martin).

SW DEVICE DRIVER

Proof of Concept/Prototype	4w	DG
Software mpeg.device	depends on prototype	

MPEG PLAYER UI

Specification and Screen Design	2w	PC/MT
Video-CD Disc Parsing	1w	DG/JH ?
Basic Player Implementation	3w	PC
Scan mode	1w	MT
Thumbnail random access	2w	MT
User testing and redesign	3w	PC/MT
Modify old CDGS player, integrate	2w	PC
Testing	2w	PC/MT

Brings us to October 15

At the end of the specification/design period, we may have a design into which the existing CDGS-Audio player doesn't fit very well. The hope would then be that the two weeks of integration would be saved and spent instead to lengthen the Basic Player Implementation step to include audio issues. This alternative might accelerate the development of the stand-alone player, too.

OTHER SW ISSUES:

Translations: It may be impossible to convey the Video Player's function set with pictograms, therefore some translation would be necessary. We should have the initial strings after the specification period. Issue: what about Korean, Japanese, etc., which are "supported" languages? (We could encode bitmaps instead of fonts for these).

Font licensing: We don't have a suitable video font. The best one I've found so far is Scala's Compact 24 and 16. These or others may need to be licensed and/or modified.

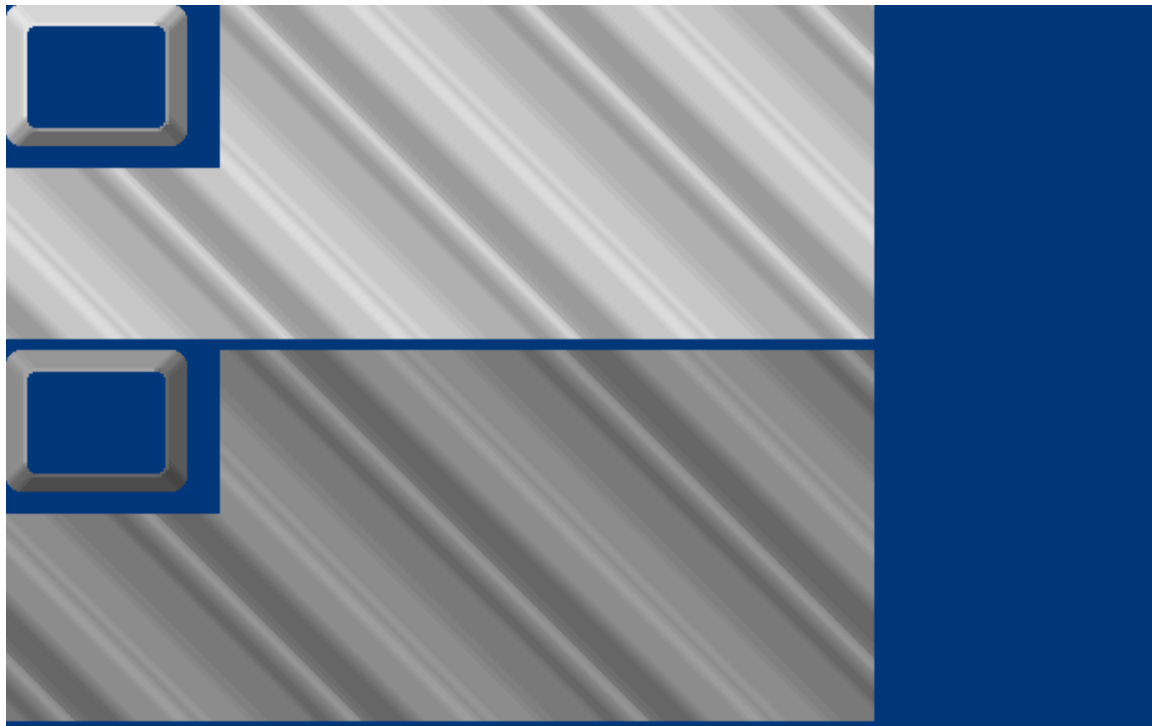
Manuals: The Video Player docs would need to be written. The Audio Player docs may need modification. Issue: final design will only be in place four weeks prior to shipping.

OTHER QUESTIONS:

Given that the focus is on CDGS/MPEG, what is the current importance of the Zorro-II version (note that the driver seems to be easy, but the stand-alone player could be more work).

chromesample:

Used as a texture file to build the chrome shapes in the actual design.



ChromeScreen:



ChromeScreen2:



ChromeScreenB:

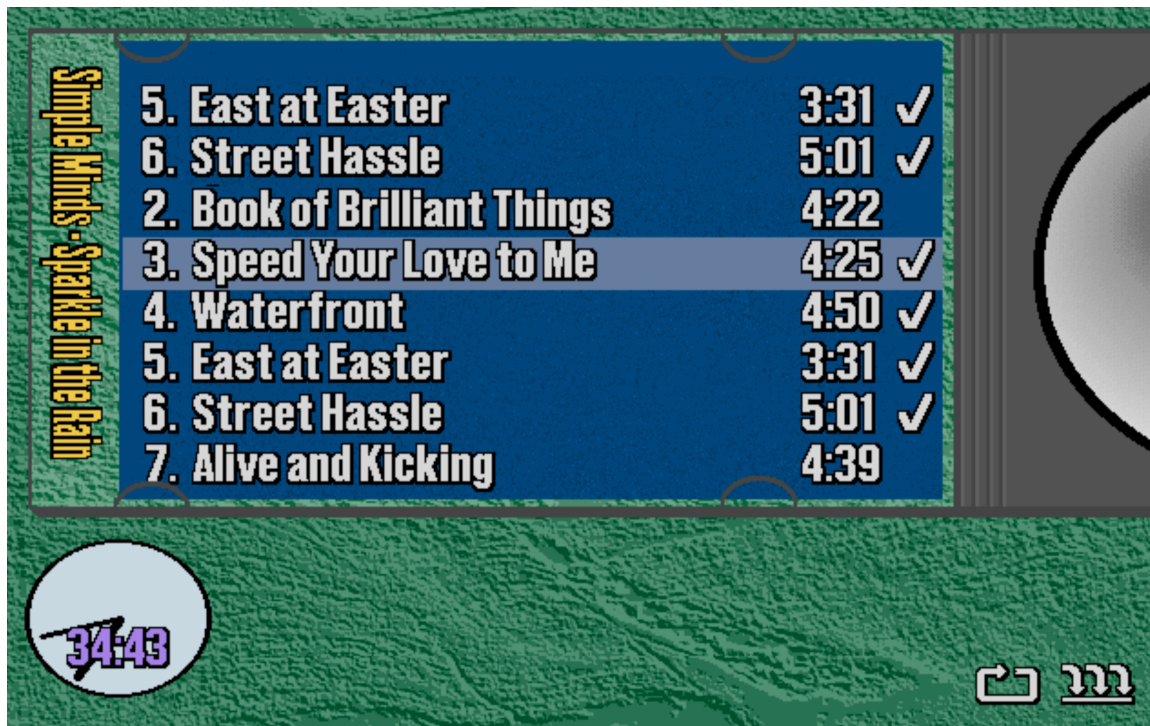


ChromeScreen2B:



Alternate designs either created by – or based on suggestions by – Martin Taillefer.

ChromeScreenMartin:



ChromeScreenMartin2:

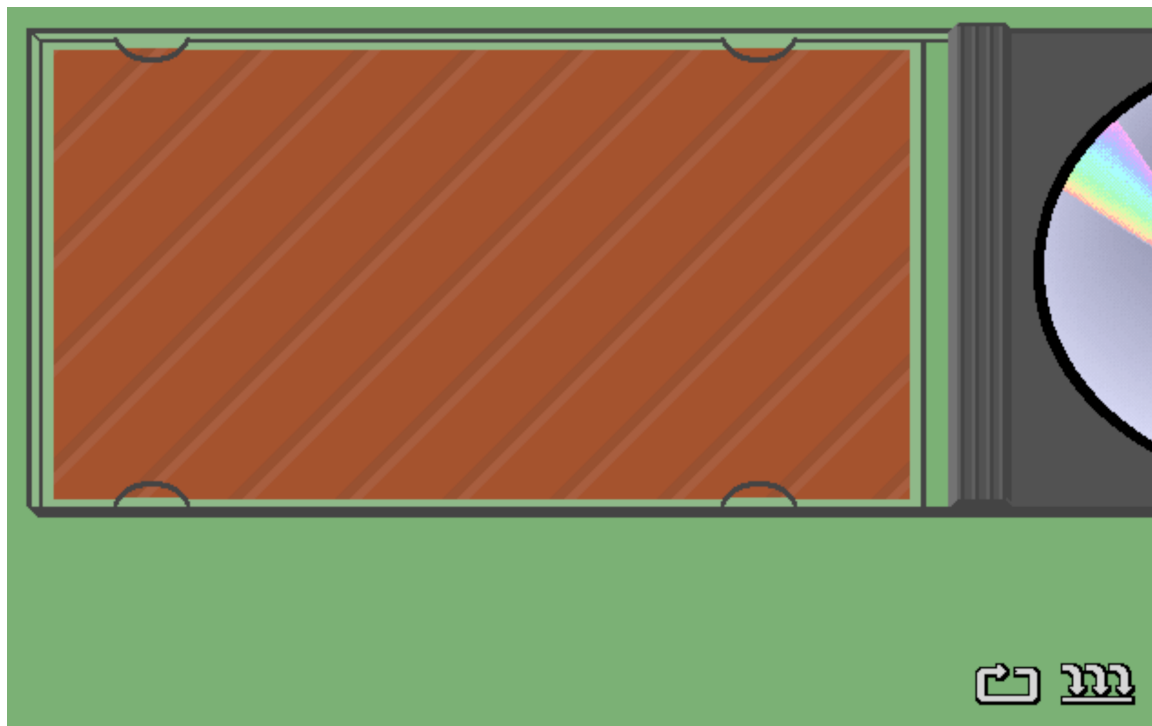


ChromeScreenNew:

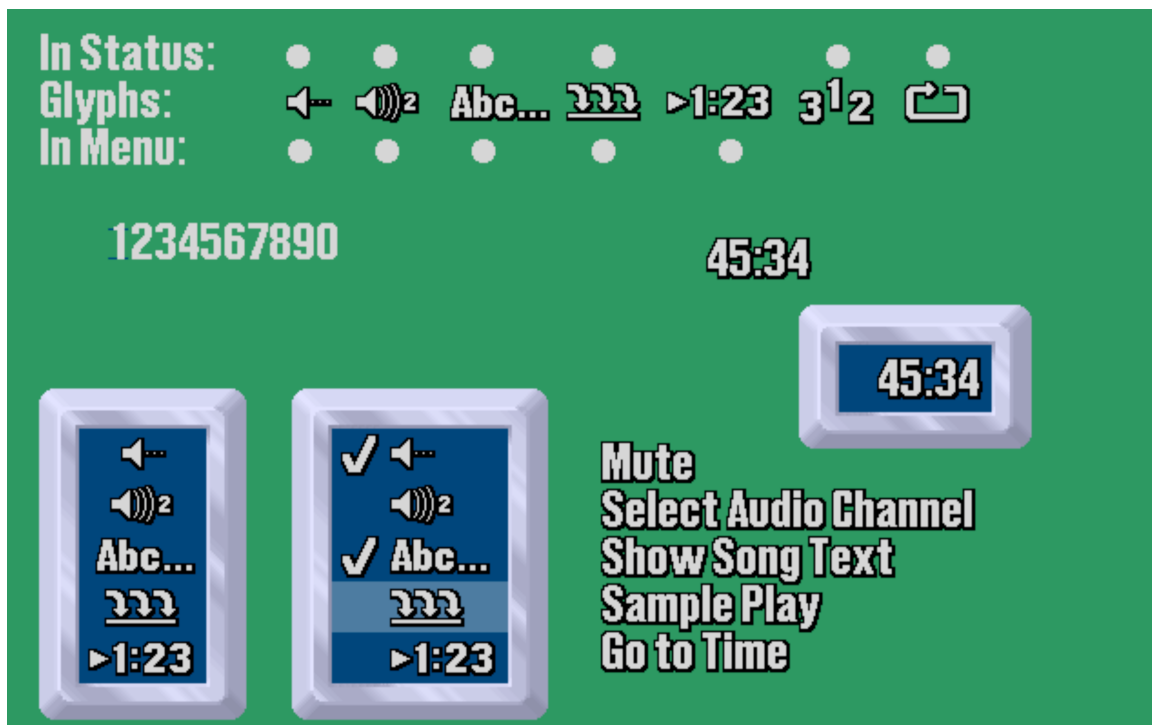


Starting point for making the screens, and elements to use / element experiments.

JewelBoxBlank:



JewelMenu:



MainScreen:



MainScreen.green:

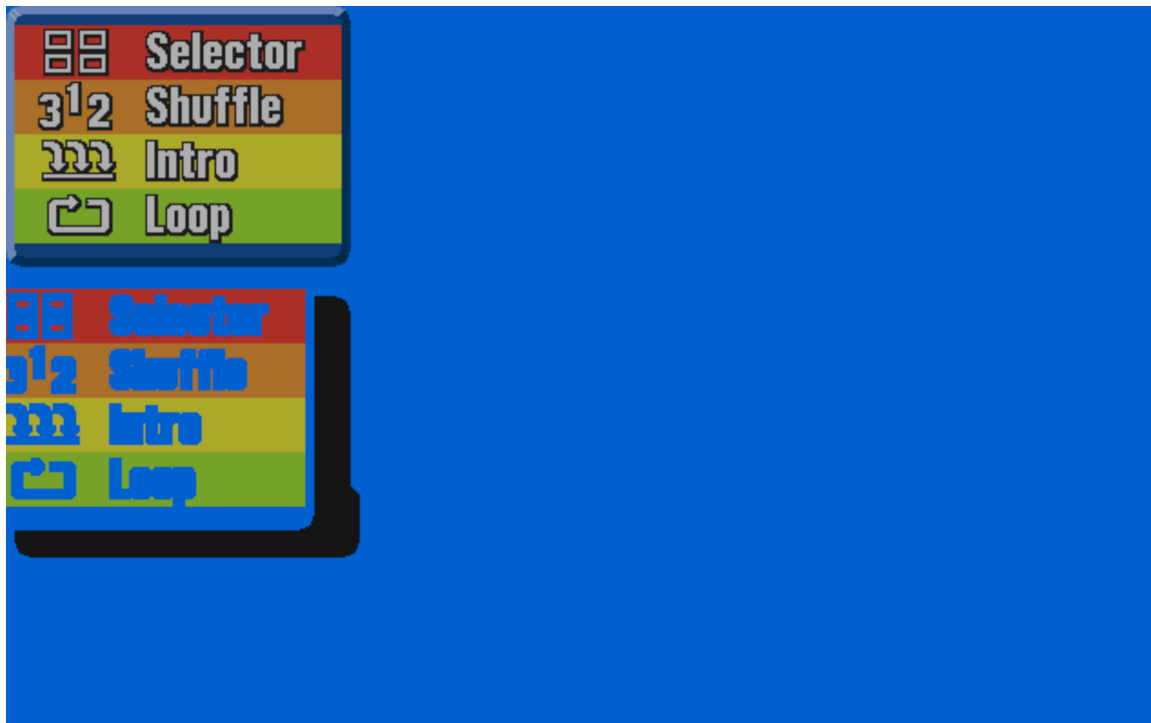


I don't recall why I had the rainbow colors. It may have been to verify that the ascenders / descenders fit, or to let me show changing the highlighted items using color cycling in DPaint.

Menu :



Menu2 :



TrackList:

1. Up on the Catwalk	4:45	✓	
2. Book of Brilliant Things	4:22	✓	
3. Speed Your Love to Me	4:25	✓	
4. Waterfront	4:50	✓	
5. East at Easter	3:31	✓	
6. Street Hassle	5:01	✓	
7. All the Things She Said	4:19	✓	
8. Alive and Kicking	2:39	✓	
9. Real Life	6:01	✓	

Selector1:

Bonnie Raitt - Nine Lives

1. No Way to Treat a Lady
2. Runnin' Back to Me
3. Who But a Fool (Thief into Paradise)
- ✓ 4. Crime of Passion
- ✓ 5. All Day, All Night
- ✓ 6. Stand Up to the Night
7. Excited
8. Freezin' (for a Little Human Love)